Goal :

* Question
  + Metric

1. Produkten skall levereras i tid

* Hur lång tid tar det att leverera
  + Tidsestimering
  + Personliga erfarenheter
* Hur mycket marginal har vi i tid?
  + Tidsestimering
  + Enkät om hur stressigt projektet är bland gruppmedlemmarna
* Hur stor del av programmet som är färdigt?
  + Andelen sprinter som är färdiga/ ännu ogjorda
  + Antalet funktioner som är färdiga / totala antalet funktioner
  + Projektchefens subjektiva bedömning

1. Produkten skall levereras med god kvalitet, med få buggar

* Hurudana processer har vi i projektet?
  + Andel av tiden som utgörs av parprogrammering
  + Antalet parallella delprojekt som utförs samtidigt
  + Antal personer per delprojekt
* Hur pass regelbundet körs tester i programmeringen?
  + Testintervall
* Hur mycket tid används till testning?
  + Tidsuppföljning
  + Testtid/programmeringstid
* Hur arbetar man inom gruppen?
  + Antal personer som arbetar
  + Antal testare mot antal programmerare
  + Antal testare som också är programmerare
* Hur bra är kvaliteten?
  + Antal buggar för tillfället per 1000 rader kod
  + Antalet kritiska buggar i systemet
  + Antal fungerande funktioner

1. Koden bör vara väldokumenterad, bör underlätta underhåll

* Hur många arbetar med projektet?
  + Antalet parallella delprojekt som utförs samtidigt
  + Antal personer per delprojekt
  + Subjektiv bedömning av programkodens dokumentation
* När sker dokumenteringen?
  + Tidsuppföljning
  + Enkät bland programmerarna
* Hur mycket dokumentation behövs?
  + Subjektiv uppskattning av programmet
  + Kravspecifikation
  + Designspecifikation
* Hur mycket dokumentation finns?
  + Antal rader dokumentation i projektet
  + Antal rader dokumentation per del
* Hur relevant är dokumentationen?
  + Subjektiv uppskattning av dokumentationen

1. Koden bör vara välskriven

* Hurudana processer har man?
  + Antal personer per grupp som arbetar med projektet
  + Antal parallella programmeringsgrupper
* Hur mycket dokumentation finns inne i koden?
  + Antal kodrader dokumentation / programkod
  + Enkät bland programmerarna
* Hur standardiserad är programkoden?
  + Subjektiv bedömning av projektchefen
  + Enkät bland programmerarna
* Hur enhetligt är programmet skrivet?
  + Stickprov från olika funktioner och subjektiv bedömning

1. Effektiviteten bör öka

* Hur effektiva är programmerarna?
  + Antalet färdiga cykler /sprint
  + Antalet buggar /sprint
* Hur bra trivs programmerarna med sina arbeten?
  + Enkät bland de anställda
* Hur bra är motivationen i projektgruppen?
  + Enkät bland de anställda
  + Subjektiv uppskattning av projektchefen

De flesta av våra mätdata i denna uppgift är interna data, som mestadels kommer att vara ostandardiserade dokument för att jämföras inom projektgruppen. Det finns dock vissa dokument som kommer att användas som mätdata som kan anses vara externa mätdata och det är i den tredje uppgiften där vi tänkte använda oss av kravspecifikationen och designspecifikationen för att få reda på hur mycket dokumentation som behövs i projektet.

I frågan om produktdata eller processdata så använder vi oss av båda delarna, men försöker i mån av möjlighet använda oss av produktdata. Processdata är aningen svårt för oss att ta in eftersom vi inte helt har bestämt ännu hur man kommer att arbeta med projektet, och det inte finns klara data att utgå ifrån.

1. Mätdata:
   * Tidsestimering
   * Personliga erfarenheter
   * Andelen sprinter som är färdiga/ ännu ogjorda
   * Antalet funktioner som är färdiga / totala antalet funktioner
   * Andel av tiden som utgörs av parprogrammering
   * Testintervall
   * Testtid/programmeringstid
   * Antal personer som arbetar på projektet
   * Antal testare mot antal programmerare
   * Antal testare som också är programmerare
   * Antal buggar för tillfället per 1000 rader kod
   * Antalet kritiska buggar i systemet
   * Antal fungerande funktioner
   * Antal personer per delprojekt
   * Subjektiv bedömning av programkodens dokumentation
   * Tidsuppföljning
   * Enkät bland programmerarna
   * Kravspecifikation
   * Designspecifikation
   * Antal rader dokumentation i projektet
   * Antal rader dokumentation per del
   * Antal parallella programmeringsgrupper
   * Antal kodrader dokumentation / programkod
   * Subjektiv bedömning av projektchefen
   * Stickprov från olika funktioner och subjektiv bedömning
   * Antalet färdiga cykler /sprint
   * Antalet buggar /sprint

Som synes här har vi relativt många olika mätdata, men de flesta av mätdatan borde vara relativt enkelt att samla in och analysera. Den största delen som kräver alla arbetares uppmärksamhet kommer i detta fall att vara enkäten som vi tänker att vi ska göra. Den kommer att ta relativt mycket tid i början, men vartefter kan data samlas in mera sporadiskt och behöver inte vara lika omfattande. Dock borde man ändå samla in information för att kunna identifiera projektets gång. I slutet av projektet skulle det kanske kunna vara ännu en större enkät som utvärderar projektet och ger oss data för att kunna förbättra våra processer och produkter i framtiden.

En annan sak som kan vara problematisk är tidsuppföljningen som vi inte är helt säkra på hur man skulle kunna göra smidigt i ett projekt som detta. Oberoende av vilket verktyg som man använder för uppföljning så kan man inte helt övervaka hur folk arbetar. Alltför noggrann tidsuppföljning kan också leda till att programmerarna i gruppen känner sig hotade och mister sin motivation att arbeta på grund av ”storebror ser dig”- känsla.

Att samla in buggar kan också vara problematiskt, eftersom man kan relativt enkelt skapa missvisande data genom att låta bli att rapportera buggar, men å andra sidan kan man inte heller belöna uppsökning av buggar då detta kan leda till att programmerarna med vilje skapar buggar som de senare löser enkelt och på så vis förbättrar sina mätdata.

Också vårt förslag att mäta antalet rader dokumentation och mäta detta kan leda till att programmerare skriver onödigt långa och omständiga kommentarer i programmet, som då samtidigt innebär att den effektiva programmeringstiden minskar och tiden satt på kommentarer ökar.